

РАЗРАБОТКА ТЕЛЕФОННОЙ КНИГИ ОРГАНИЗАЦИИ ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID

Аннотация

В статье рассматривался метод создания модели сервера приложений на примере разработки телефонной книги организации для ОС Android с использованием СУБД MySQL и API сервера на NodeJS.

Ключевые слова: информационные технологии, ИТ-проект, информационные системы, Android, MySQL, NodeJS, телефонная книга.

Abstract

This article discusses a method of creating model application server for an example of development phone book of organization for Android using the MySQL and NodeJS API server.

Keywords: Information Technology, IT project, Information Systems, Android, MySQL, NodeJS, phone book.

В наше время смартфоны и прочие мобильные устройства стали неотъемлемой частью жизни каждого. Помимо стандартного свойства совершать звонки смартфоны имеют ряд других возможностей. Благодаря большой вычислительной мощности, которую имеют смартфоны сегодня, стало возможным устанавливать на них более развитую операционную систему, открытой для разработки программного обеспечения сторонними разработчиками (операционная система обычных мобильных телефонов закрыта для сторонних разработчиков) [1], что позволило создать огромное множество различных приложений. Таким образом, смартфон стал похож на ПК в плане возможностей, но при этом остался все тем же устройством для совершения звонков, которое можно носить в кармане.

Поэтому, было решено написать приложение для операционной системы Android, с помощью которого можно быстро найти контакты любого сотрудника компании и сделать вызов, или отправить e-mail. При этом, контакты не должны занимать память телефона, а должны подгружаться с сервера. Это освободит пользователя от ручного обновления и добавления контактов.

Для этого была выбрана модель сервера приложений (рисунок 1).

Необходимо предусмотреть то, что создаваемый сервис должен быть создан для нескольких компаний-заказчиков. Поэтому, сервер должен быть доступен повсеместно, иметь хорошую скорость соединения и высокую безотказность. В следствие чего, был выбран сервис AWS. Сервис AWS может предоставить огромную вычислительную мощность за небольшую цену. С помощью этого сервиса можно создать сервер для API (EC2) и сервер для БД (RDS). В качестве СУБД был выбран MySQL. Обмен данными между серверной частью и клиентской должен осуществляться посредством HTTPS [2] протокола. Это необходимо для защиты передаваемой информации от злоумышленников.

БД спроектирована таким образом, чтобы пользователь одной компании не видел пользователей другой (рисунок 2).

API [3] сервер должен делиться на модули. Это необходимо для обеспечения расширяемости и читаемости кода (рисунок 3). В качестве языка, на котором будет написано API, предлагается выбрать NodeJS - платформа, основанная на JavaScript движке V8, и предлагающая асинхронное API для работы с сетью и диском [4].

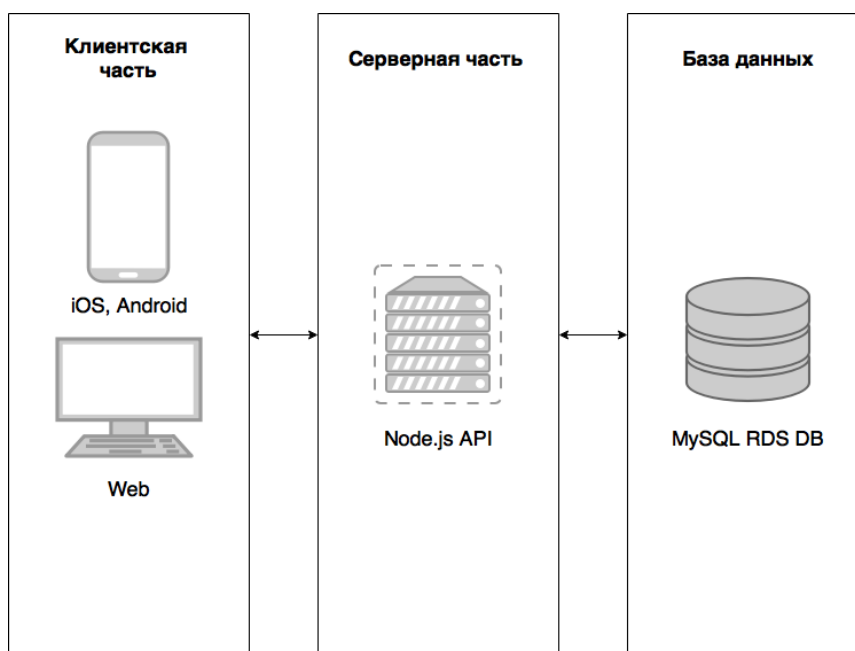


Рис. 1. Модель сервера приложений

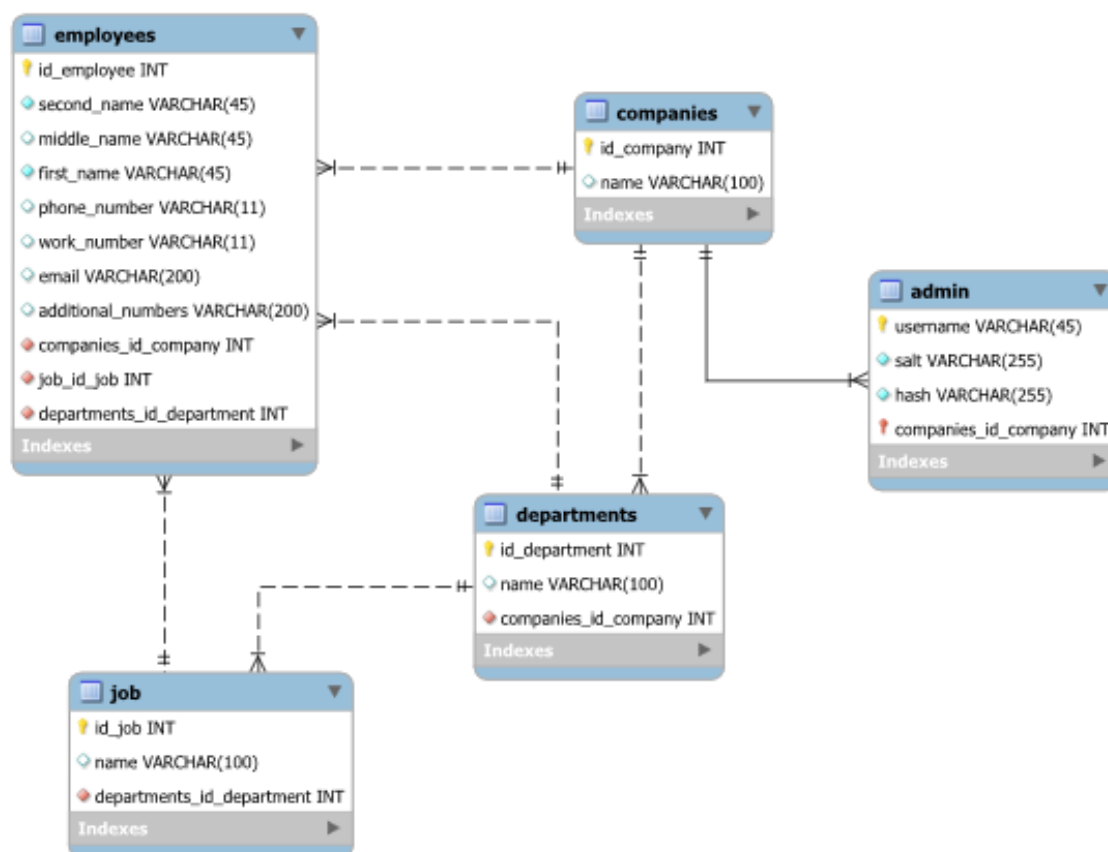


Рис. 2. Схема БД

API

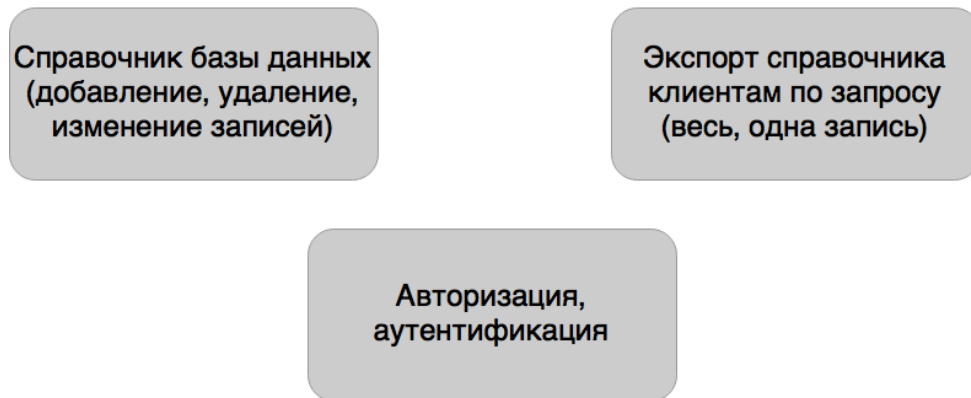


Рис. 3. Схема API

Приложение так же можно разделить на несколько модулей (рисунок 4).

Клиент

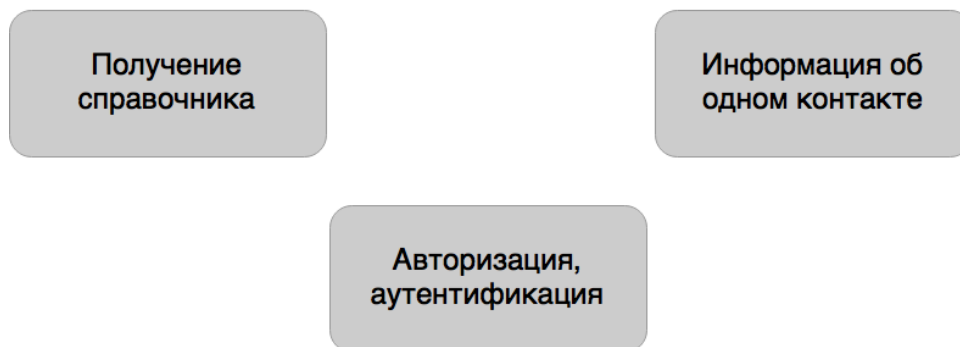


Рис. 4. Схема приложения

Данная тема актуальна для компаний с большим количеством сотрудников. Возможность быстро связаться с любым из сотрудников позволяет оптимизировать общение между людьми. Особенно это необходимо в экстренных ситуациях. Помимо этого, возможности данного приложения можно расширить, улучшая взаимодействие между сотрудниками.

Список использованных источников

1. Смартфон [Сайт]: URL: <https://ru.wikipedia.org/wiki/%D0%A1%D0%BC%D0%B0%D1%80%D1%82%D1%84%D0%BE%D0%BD> Дата обращения: 16.04.2016.
2. HTTPS [Сайт] URL: <https://ru.wikipedia.org/wiki/HTTPS> Дата обращения: 16.04.2016.
3. API [Сайт] URL: <https://ru.wikipedia.org/wiki/API> Дата обращения: 16.04.2016.
4. NodeJS [Сайт] URL: <http://nodejs.ru/> Дата обращения: 16.04.2016.